(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0090860
(43) 공개일자 2021년07월21일

(51) 국제특허분류(Int. Cl.)
G06T 1/20 (2018.01) *G06T 15/06* (2011.01)
G06T 15/08 (2011.01)
(52) CPC특허분류
G06T 1/20 (2013.01)
G06T 15/06 (2013.01)
(21) 출원번호 10-2020-0004110
(22) 출원일자 2020년01월13일
심사청구일자 2020년01월13일

(71) 출원인
한국기술교육대학교 산학협력단
충청남도 천안시 동남구 병천면 충절로 1600 (한국기술교육대학교내)
(72) 발명자
김덕수
대전광역시 유성구 문화원로 133, 1006호(봉명동, 스톤빌)
이지훈
전라북도 완주군 이서면 이문길 15, 101동 401호 (부영아파트)
(74) 대리인
특허법인오암

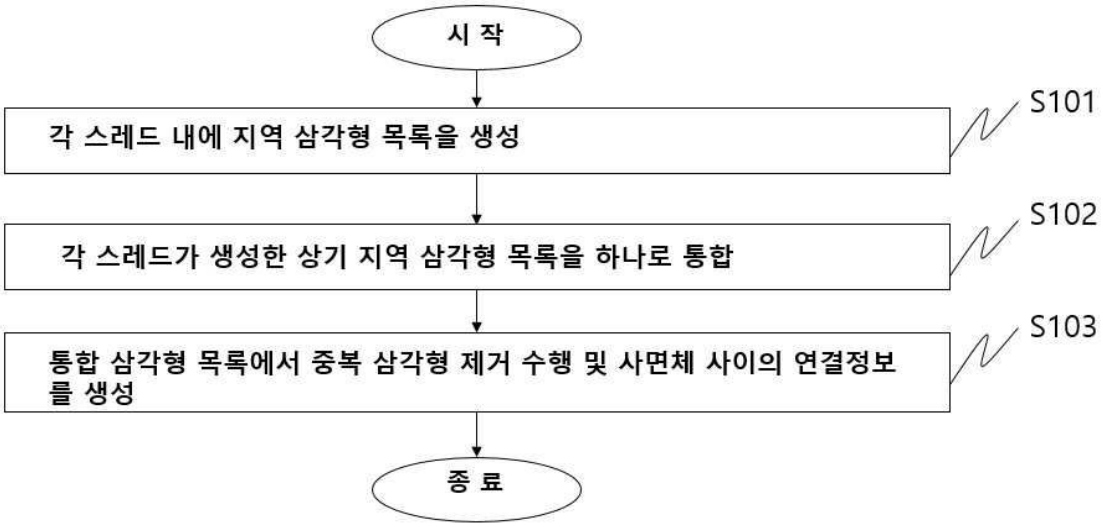
전체 청구항 수 : 총 8 항

(54) 발명의 명칭 **비정렬 격자 데이터에 대한 광선 투사를 위한 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법**

(57) 요 약

본 발명의 일 실시예에 따른 병렬 처리에 의한 연결정보 추출 알고리즘 수행방법은, 비정렬 격자 데이터에 대한 광선 투사를 위해 셀 사이의 연결정보를 추출하는 연결정보 추출 알고리즘 수행 방법에 있어서, 다중 코어를 가지는 연산장치 내 각 스레드는 할당받은 사면체를 하나씩 방문하고, 각 사면체를 구성하고 있는 네 개의 삼각형 정보를 생성하도록 각 스레드 내에 지역 삼각형 목록을 생성하는 단계; 상기 각 스레드가 생성한 상기 지역 삼각형 목록을 하나로 통합하는 단계; 상기 통합 단계에 의해 생성된 통합 삼각형 목록에서 중복 삼각형 제거 수행 및 사면체 사이의 연결정보를 생성하는 단계를 포함한다.

대 표 도 - 도1



(52) CPC특허분류
G06T 15/08 (2013.01)

명 세 서

청구범위

청구항 1

비정렬 격자 데이터에 대한 광선 투사를 위해 셀 사이의 연결정보를 추출하는 연결정보 추출 알고리즘 수행 방법에 있어서,

다중 코어를 가지는 연산장치 내 각 스레드는 할당받은 사면체를 하나씩 방문하고, 각 사면체를 구성하고 있는 네 개의 삼각형 정보를 생성하도록 각 스레드 내에 지역 삼각형 목록을 생성하는 단계;

상기 각 스레드가 생성한 상기 지역 삼각형 목록을 하나로 통합하는 단계;

상기 지역 삼각형 목록을 통합하는 단계에 의해 생성된 통합 삼각형 목록에서 중복 삼각형 제거 수행 및 사면체 사이의 연결정보를 생성하는 단계

를 포함하는 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법.

청구항 2

제1항에 있어서,

상기 각 스레드 내에 지역 삼각형 목록을 생성하는 단계는,

상기 각 스레드가 자신만의 삼각형 목록을 생성하되,

일정한 크기의 메모리 공간을 미리 잡아 둔 후, 요청이 들어오면 그 중 일부를 배분해주는 메모리 풀 기법을 이용하여 생성하는 것을 특징으로 하는 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법.

청구항 3

제1항에 있어서,

각 스레드가 생성한 상기 지역 삼각형 목록을 하나로 통합하는 단계는,

상기 지역 삼각형 목록이 해시 테이블 형태로 이루어져, 해시 키 단위로 통합을 수행하는 것을 특징으로 하는 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법.

청구항 4

제3항에 있어서,

상기 해시 키 단위로 해시 테이블 통합을 위해, 각 해시 키마다 별도의 연결리스트를 가지는 형태의 자료구조를 사용하며,

하나의 스레드에 의해 하나의 연결리스트를 다른 연결리스트에 연결하는 방식의 직렬 처리를 수행하는 것을 특징으로 하는 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법.

청구항 5

제1항에 있어서,

상기 통합 삼각형 목록에서 중복 삼각형 제거 및 사면체 사이의 연결정보를 생성하는 단계는,

상기 통합 삼각형 목록을 해시 키 단위로 각 스레드에 균등 분배하고, 상기 각 스레드가 자신에게 할당된 각 해시 키들에 대해 중복 삼각형 제거작업을 수행하는 것을 특징으로 하는 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법.

청구항 6

제5항에 있어서,

상기 중복 삼각형 제거작업은,

삼각형을 순차적으로 방문하며 중복된 삼각형을 찾는 방법으로 수행하되,

하나의 삼각형을 찾으면 탐색이 종료되며, 다음 삼각형에 대한 중복 삼각형 제거 작업으로 넘어가는 것을 특징으로 하는 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법.

청구항 7

제6항에 있어서,

상기 중복된 삼각형을 제거할 때는

해당 삼각형을 포함하는 사면체의 삼각형 구성 정보를 갱신하여, 사면체들 사이의 연결정보가 생성하며,

서로 다른 해시 키를 가지는 삼각형들 사이에 중복 검사를 수행하지 않도록 하여 서로 독립적으로 병렬 수행 가능한 것을 특징으로 하는 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법.

청구항 8

제1항 내지 제7항 중 어느 한 항의 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법을 수행하며, 컴퓨터가 판독 가능한 기록 매체에 저장하는 컴퓨터 프로그램.

발명의 설명

기 술 분 야

[0001] 본 발명은 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법에 관한 것으로, 더욱 상세하게는 비정렬 격자에 대한 광선투사 수행의 전처리 과정 중 하나인 셀 사이 연결정보 추출에 대한 병렬처리 알고리즘 수행 방법에 관한 것이다.

배 경 기 술

[0002] 직접 볼륨 렌더링(Direct Volume Rendering, DVR)은 대표적인 과학적 가시화 기법 중 하나로, M&S, 의료 등 다양한 분야에서 폭 넓게 활용된다. 하지만 비정렬 격자의 경우 데이터의 분포가 불규칙적으로, 가시화 연산에 많은 시간이 소요된다. 최근 많은 응용분야에서 비정렬 격자의 사용이 증가하고 있으며, 비정렬 격자에 대한 정확하면서도 빠른 볼륨 렌더링 알고리즘에 대한 요구가 커져가고 있다.

[0003] 광선 투사(ray casting)는 이미지 평면의 각 픽셀(pixel)로 쏘아진 광선의 횡단(ray traversal)을 따라가며, 광선이 만나는 셀(cell)들의 값에 따라 픽셀의 색상을 계산하는 직접 볼륨 렌더링(이후, 볼륨 렌더링) 기법으로, 볼륨 렌더링 기법중 가장 정확한 결과를 생성한다. 하지만 비정렬 격자의 경우, 비정렬 격자를 구성하는 셀들의 기하학적 정보를 활용하여 광선이 지나는 셀들을 찾아야 한다. 이를 위해 셀 사이 연결정보(connectivity information)가 필요하며, 이는 셀을 구성하는 다면체들 사이의 면(face) 공유 정보로 표현된다. 따라서 다면체들을 구성하는 면(예, 삼각형) 정보 및 그들 사이의 공유 정보를 추출하는 전처리 과정이 필요하다.

[0004] 비정렬 격자에 대한 광선 투사는 연산양이 많아 대용량 데이터에 적용하기 어렵다는 한계를 가지며, 이를 극복하기 위해 다양한 가속화 기법 및 병렬처리 알고리즘들이 발명되어 왔다. 하지만 기존 발명들은 대부분 광선 횡단 자체의 성능을 개선하는 것에 초점을 맞추고 있으며, 전처리 단계의 성능 향상에 대한 발명은 찾기 힘들다.

[0005] 이처럼 비정렬 격자에 대한 광선 투사 성능 향상을 위해 멀티코어 CPU 또는 GPU 기반의 다양한 병렬처리 기술들이 활발히 발명되어 왔다. 하지만, 기존 발명들은 대부분 광선횡단 단계의 병렬 처리에 집중하고 있으며, 전처리 단계에 대한 고민은 미미하다.

[0006] 전처리 단계 성능 개선을 위한, 비정렬 격자에 대한 광선 횡단 알고리즘 및 기존의 연결정보 추출 알고리즘에 대해서 설명하면 다음과 같다.

[0007] 광선 횡단의 첫 단계는 각각의 픽셀로 쏘아지는 광선에 대한 진입 면(entry face)을 찾는 것이다. 이는 하나의 셀에만 속하는(외부와 만나는) 외곽 면(boundary face)을 이미지 평면에 투영(projection)하는 방법으로 찾는다. 비 볼록 다면체 메시의 경우, 한 픽셀로 투영된 외곽 면들을 깊이 값(depth 또는 Z-value)을 기준으로

정렬함으로써, 진입 순서를 결정할 수 있다. 그 중 가장 작은 깊이 값을 가지는 면이 해당 픽셀로 쏘아지는 광선의 최초 진입 면이 된다.

[0008] 광선의 횡단은 진입 면에서부터 시작된다. 즉, 진입 면이 속한 셀이 광선이 처음 만나는 셀이며, 이후 광선 투사는 다음 과정을 통해 진행된다. 1) 현재 셀(다면체)을 구성하는 나머지 면들 중, 광선이 지나는 면을 찾는다. 2) 한 면은 두 개 셀(외곽 면인 경우, 한 개 셀)에 의해 공유되므로, 해당 면을 공유하는 셀 중 현재 셀이 아닌 셀이 광선이 다음에 지나는 셀이 된다. 3) 위 1~2 과정을 다음 지나는 면이 외곽 면이 아닐 때 까지 반복한다. 4) 광선이 외곽 면을 만난다는 것은 광선이 비정렬 격자 외부로 나왔다는 것을 뜻하며, 다음 진입 면으로 진입하여 위 과정을 반복한다.

[0009] 상술한 바와 같이, 비정렬 격자에 대한 광선 횡단 혹은 투사를 위해서는 셀 사이의 연결정보가 필요하다. 비정렬 격자 데이터는 각 셀의 다면체를 구성하는 정점(vertex) 정보를 가지고 있다. 일반적으로 사면체(tetrahedral)를 많이 사용하며, 네 개의 정점 정보로 표현된다. 따라서 광선 투사 수행을 위해서는 사면체를 구성하는 정점들로부터, 사면체들 사이의 연결정보를 추출하는 과정이 필요하다.

[0010] 정점 정보로 구성된 사면체 메시(mesh)의 연결정보 추출방법에 대한 문헌을 찾기 어렵기 때문에, 본 발명에서는 과학적 가시화를 위해 널리 사용되는 오픈소스 라이브러리인 VTK(Visualization ToolKit)의 구현을 기준으로 기존 연결정보 추출 알고리즘을 설명한다.

[0011] VTK는 효율적인 광선 횡단 구현을 위해, 사면체 메시의 연결정보를 사면체들을 구성하는 삼각형 목록과 사면체 사이의 삼각형 공유 정보로 표현한다. 이를 위해서는 사면체의 정점 정보로부터 삼각형 목록을 생성하는 과정이 필요하다. 한 사면체를 구성하는 네 개의 삼각형은 네 정점을 세 개씩 묶는 방법으로 간단히 생성할 수 있다. 하지만, 인접한 두 사면체는 하나의 삼각형을 공유하므로, 이 과정에서 중복된 삼각형이 생성될 수 있다. VTK의 구현은 삼각형들의 정보를 중복 없이 저장하고, 각 사면체에는 사면체를 구성하는 삼각형의 번호를 저장한다. 또한 각 삼각형에 자신이 속한 두 개의 사면체의 번호를 기록하는 방법으로 셀 사이 연결정보를 생성한다.

[0012] 중복되는 삼각형 정보 없이 연결정보를 추출하는 VTK 알고리즘은 다음과 같다. 1) 사면체를 순차적으로 방문하면서, 2) 해당 사면체에 속하는 삼각형들을 하나씩 확인한다. 3) 해당 삼각형을 구성하는 정점들을 번호(vertex ID)를 기준으로 정렬한 후, 동일한 정점들로 구성된 삼각형이 있는지 확인한다. 4-1) 만약 해당 삼각형이 존재하면, 해당 삼각형의의 번호를 사면체에 기록한다. 4-2) 해당 삼각형이 존재하지 않는 경우, 새로운 삼각형 정보를 생성하고 다면체에 기록한다. 5) 모든 사면체를 방문할 때까지, 1)~4)과정을 반복한다.

[0013] 이러한 과정을 통해 중복된 삼각형 정보 없이, 사면체 메시의 연결정보를 추출할 수 있다. 하지만, 모든 삼각형마다 기존에 중복된 삼각형이 있는지 여부를 검사해야 하며, 이는 많은 연산 시간을 소요하는 작업이다. 중복 검사 시 탐색 공간을 줄이기 위해 삼각형을 구성하는 가장 낮은 정점 번호를 키(Key)로 사용하는 해시 알고리즘을 사용한다. 하지만, 비정렬 격자 데이터의 크기(격자(다면체)의 수)가 커짐에 따라, 해시 테이블(hash table)의 크기가 증가하고 탐색에 걸리는 시간이 급격이 증가한다는 한계점을 가진다.

[0014] 따라서, 전술한 문제를 해결하기 위하여 면 정보 및 연결정보를 추출하는 전처리 단계를 위한 병렬처리 알고리즘을 적용하고, 셀 사이 연결정보를 사용하는 광선 투사 시스템에 공통적으로 적용할 수 있는 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법에 대한 발명이 필요하게 되었다.

선행기술문헌

특허문헌

[0015] (특허문헌 0001) (비특허문헌 001) K. Brodlie, J. Wood, "Recent advances in volume visualization", Computer Graphics Forum, Vol. 20, No.2, 2001.

(특허문헌 0002) (비특허문헌 002) M. Garrity. "Raytracing irregular volume data", Computer Graphics, 1990.

(특허문헌 0003) (비특허문헌 003) Will Schroeder, Kenneth M. Martin, and William E.Lorensen, "The visualization toolkit (4th ed.): An object-oriented approach to 3D graphics", Prentice-Hall, Inc., 2006.

발명의 내용

해결하려는 과제

[0016] 본 발명은 비정렬 격자 데이터에 대한 광선 투사 수행을 위해 필요한 셀 사이 연결정보 추출 성능 개선을 위한 병렬처리 알고리즘을 수행하는 방법을 제공하는데 목적이 있다.

과제의 해결 수단

[0017] 본 발명의 일 실시예에 따른 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법은, 비정렬 격자 데이터에 대한 광선 투사를 위해 셀 사이의 연결정보를 추출하는 연결정보 추출 알고리즘 수행 방법에 있어서, 다중 코어를 가지는 연산장치 내 각 스레드는 할당받은 사면체를 하나씩 방문하고, 각 사면체를 구성하고 있는 네 개의 삼각형 정보를 생성하도록 각 스레드 내에 지역 삼각형 목록을 생성하는 단계; 상기 각 스레드가 생성한 상기 지역 삼각형 목록을 하나로 통합하는 단계; 상기 통합 단계에 의해 생성된 통합 삼각형 목록에서 중복 삼각형 제거 수행 및 사면체 사이의 연결정보를 생성하는 단계를 포함한다.

[0018] 상기에 있어서, 상기 각 스레드 내에 지역 삼각형 목록을 생성하는 단계는, 상기 각 스레드가 자신만의 삼각형 목록을 생성하되, 일정한 크기의 메모리 공간을 미리 잡아 둔 후, 요청이 들어오면 그 중 일부를 배분해주는 메모리 풀 기법을 이용하여 생성하는 것을 특징으로 한다.

[0019] 상기에 있어서, 각 스레드가 생성한 상기 지역 삼각형 목록을 하나로 통합하는 단계는, 상기 지역 삼각형 목록이 해시 테이블 형태로 이루어져, 해시 키(hash key) 단위로 통합을 수행하는 것을 특징으로 한다.

[0020] 상기에 있어서, 상기 해시 키 단위로 해시 테이블 통합을 위해, 각 해시 키 마다 별도의 연결리스트(linked list)를 가지는 형태의 자료구조를 사용하며, 하나의 스레드에 의해 하나의 연결리스트를 다른 연결리스트에 연결하는 방식의 직렬 처리를 수행하는 것을 특징으로 한다.

[0021] 상기에 있어서, 상기 통합 삼각형 목록에서 중복 삼각형 제거 및 사면체 사이의 연결정보를 생성하는 단계는, 상기 통합 삼각형 목록을 해시 키 단위로 각 스레드에 균등 분배하고, 상기 각 스레드가 자신에게 할당된 각 해시 키들에 대해 중복 삼각형 제거작업을 수행하는 것을 특징으로 한다.

[0022] 상기에 있어서, 상기 중복 삼각형 제거작업은, 삼각형을 순차적으로 방문하며 중복된 삼각형을 찾는 방법으로 수행하되, 하나의 삼각형을 찾으면 탐색이 종료되며, 다음 삼각형에 대한 중복 삼각형 제거 작업으로 넘어가는 것을 특징으로 한다.

[0023] 상기에 있어서, 상기 중복된 삼각형을 제거할 때는 해당 삼각형을 포함하는 사면체의 삼각형 구성 정보를 갱신하여, 사면체들 사이의 연결정보가 생성하며, 서로 다른 해시 키를 가지는 삼각형들 사이에 중복 검사를 수행하지 않도록 하여 서로 독립적으로 병렬 수행 가능한 것을 특징으로 한다.

발명의 효과

[0024] 본 발명의 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법은 병렬처리 알고리즘에 의해 각 단계에서 스레드들이 독립적으로 작업을 수행함으로서, 높은 병렬처리 효율을 얻는다는 장점이 있다.

[0025] 또한, 중복 검사 과정에서 메모리 접근의 공간적 지역성을 높임으로서 높은 캐시 활용 효율을 얻을 수 있다.

[0026] 또한, 기존의 직렬 처리 알고리즘을 단순히 병렬화했을 때 발생하는 처리 과정의 종속성(동기화)을 해결하고, 직렬 처리 알고리즘 대비 32개의 스레드(16개 물리코어)를 활용하여 더 높은 성능을 보여주는 장점이 있다.

도면의 간단한 설명

[0027] 도 1은 본 발명의 일 실시예에 따른 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법의 순서도이다.

도 2는 본 발명의 셀 연결 정보 추출 병렬 처리 알고리즘을 개략적으로 설명하기 위한 도면이다.

도 3은 본 발명에서 삼각형 목록 (h : # 해시 키 수)의 데이터 구조와 전체 삼각형 목록을 결합하는 방법을 보여주는 도면이다.

도 4는 세 가지 서로 다른 벤치 마크 데이터 세트의 볼륨 렌더링 이미지 결과 예이다.

도 5a 내지 도 5c는 본 발명에서 컴퓨팅 스레드 수에 따른 병렬 처리 알고리즘의 성능들을 보여주는 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0028] 이하에서는 도면을 참조하여 본 발명의 구체적인 실시예를 상세하게 설명한다. 다만, 본 발명의 사상은 제시되는 실시예에 제한되지 아니하고, 본 발명의 사상을 이해하는 당업자는 동일한 사상의 범위 내에서 다른 구성요소를 추가, 변경, 삭제 등을 통하여, 퇴보적인 다른 발명이나 본 발명 사상의 범위 내에 포함되는 다른 실시예를 용이하게 제안할 수 있을 것이나, 이 또한 본원 발명 사상 범위 내에 포함된다고 할 것이다.

[0029] 기존 기술에서 살펴본 바와 같이, 사면체들의 연결정보를 추출하는 과정은 많은 연산을 요구한다. 본 발명은 최근 대부분의 연산 장치들이 여러 개의 CPU 코어를 가진다는 점에 착안하여, 병렬처리에 의한 연결정보 추출 알고리즘을 제안한다.

[0030] 기존의 VTK의 연결정보 추출 알고리즘을 병렬처리 알고리즘으로 확장하는 직관적인 접근법은 사면체들을 스레드(thread)들에게 균등분배하고, 각 스레드는 기존 방법을 사용하여 할당 받은 사면체들 사이의 연결정보를 추출하는 것이다.

[0031] 하지만, 이 방법은 서로 다른 스레드로 분배된 사면체들 사이의 연결정보를 얻을 수 없다는 문제점을 가진다. 놓치는 정보 없이 정확한 연결정보를 추출하기 위해서는, 삼각형의 중복 확인(VTK의 연결정보 추출 알고리즘에서 동일한 정점들로 구성된 삼각형이 있는지 확인하는 단계) 단계에서 다른 스레드들이 생성한 삼각형 목록들을 모두 확인해야 한다.

[0032] 즉, 삼각형 중복 검사 시마다 다른 스레드들과 동기화 작업이 필요하며, 이는 사면체 하나를 처리할 때 마다 네 번씩 발생하는 매우 빈번한 작업이다. 동기화(synchronization)는 병렬처리의 효율을 떨어뜨리는 주요 병목 지점으로, 이처럼 잦은 동기화는 병렬처리가 직렬처리보다 오히려 낮은 성능을 보이는 원인이 되기도 한다. 실제로, 위의 간단한 병렬처리 알고리즘을 구현해 본 결과, 8개 스레드를 사용하는 경우 본 발명에 사용된 벤치마크 데이터들에 대해, 직렬처리 대비 최대 6배(평균 3.2배) 낮은 성능을 보여주었다.

[0033] 본 발명은 스레드들의 사이의 동기화를 제거하고 높은 병렬처리 효율을 얻기 위해, 하기의 3-단계로 이루어진 병렬처리에 의한 연결정보 추출 알고리즘을 제안한다.

[0034] 도 1은 본 발명의 일 실시예에 따른 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법의 순서도이며, 도 2는 본 발명의 셀 연결 정보 추출 병렬 처리 알고리즘을 개략적으로 설명하기 위한 도면이며, 도 3은 본 발명에서 삼각형 목록 (h : # 해시 키 수)의 데이터 구조와 전체 삼각형 목록을 결합하는 방법을 보여주는 도면이다.

[0035] 도 1 내지 도 3을 참조하면, 3-단계로 이루어진 병렬처리에 의한 연결정보 추출 알고리즘은 다음과 같이 수행된다.

[0036] 첫 번째 단계는, 스레드 내에 지역 삼각형 목록(local triangle list)을 생성하는 단계다(S101).

[0037] 사면체들은 스레드들에게 균등하게 되며, 스레드들은 서로 동기화없이 작업을 수행한다. 각 스레드는 할당받은 사면체를 하나씩 방문하며 각 사면체를 구성하고 있는 네 개의 삼각형 정보를 생성한다.

[0038] 이 과정에서 각 스레드는 자신만의 삼각형 목록을 만들며, 삼각형 목록은 기존 직렬 알고리즘과 같이 해시 테이블 형태로 생성된다.

[0039] 하지만 기존 직렬 알고리즘과 달리 삼각형 중복 검사를 수행하지 않으며, 지역 삼각형 목록 안에 또는 다른 스레드가 생성한 삼각형 목록에 중복된 삼각형이 존재하게 된다. 즉, 사면체들 사이의 연결정보는 아직 생성되지 않은 상태로, 다음 단계들을 통해 연결 정보가 만들어진다.

[0040] 두 번째는, 각 스레드가 생성한 지역 삼각형 목록을 하나로 통합하는 단계다(S102). 각 지역 삼각형 목록은 해시 테이블 형태로, 해시 키(hash key) 단위로 해서 하나로 통합을 수행한다.

[0041] 본 발명은 효율적인 해시 테이블 통합을 위해, 각 해시 키 마다 별도의 연결리스트(linked list)를 가지는 형태의 자료구조를 사용한다.

[0042] 따라서 하나의 연결리스트를 다른 연결리스트에 연결하는 방법으로 해시 테이블을 간단히 통합할 수 있다(도 3의 녹색 줄). 삼각형 목록 통합은 하나의 스레드에 의해 직렬로 처리된다.

[0043] 마지막 단계는, 중복 삼각형 제거 및 사면체 사이의 연결정보(cell connectivity information)를 생성하는 단계다(S103).

[0044] 본 단계에서는 두 번째 단계에서 통합 및 생성된 통합 삼각형 목록(global triangle list)을 해시 키 단위로 스레드들에게 균등 분배한다.

[0045] 스레드들은 자신에게 할당된 각 해시 키들에 대해 중복 삼각형 제거작업을 수행한다. 외부로 노출된 외곽 면을 제외한 모든 삼각형은 통합된 통합 삼각형 목록에 두 개씩 존재하게 된다.

[0046] 중복 삼각형 제거 작업은 삼각형을 순차적으로 방문하며, 중복된 삼각형을 찾는 방법으로 수행된다. 탐색은 하나의 삼각형을 찾으면 종료되며, 다음 삼각형에 대한 중복 삼각형 제거 작업으로 넘어간다.

[0047] 또한, 중복된 삼각형을 제거할 때는 해당 삼각형을 포함하는 사면체의 삼각형 구성 정보를 갱신하여, 사면체들 사이의 연결정보가 생성한다.

[0048] 서로 다른 해시 키를 가지는 삼각형들 사이에는 중복 검사가 필요 없으며, 상술한 중복 삼각형 제거 작업은 서로 독립적으로 수행가능하다. 따라서 모든 스레드들은 동기화없이 주어진 삼각형 중복 제거 작업을 독립적으로 처리할 수 있다.

[0049] 본 발명에서는 효율적인 메모리 관리를 위해 삼각형 목록을 생성하는 과정에서 연결리스트를 사용하고 메모리 풀(memory pool) 기법을 이용한다.

[0050] 새로운 삼각형 추가 시 동적 메모리 할당이 필요하지만, 동적 메모리 할당 요청들은 직렬화되어 처리되기 때문에 병렬처리 알고리즘의 성능 저하 요인이 될 수 있다.

[0051] 따라서, 본 발명은 이러한 문제를 해결하기 위해, 메모리 풀(memory pool) 기법을 이용한다. 즉, 일정한 크기의 메모리 공간을 미리 잡아 둔 후, 요청이 들어오면 그 중 일부를 배분해주는 방법을 이용한다.

[0052] 이는 추가적인 동적 할당 과정을 피할 수 있으며, 직렬화에 따른 성능 저하를 예방할 수 있다. 본 발명에서 제안하는 병렬처리 알고리즘은 스레드 별 지역 삼각형 목록을 생성하며, 그 과정에서 중복된 삼각형을 생성한다.

[0053] 하나의 면은 최대 두 개의 사면체에 의해 공유될 수 있으며, 최악의 경우 직렬처리 대비 2배의 메모리가 필요하다는 것을 의미한다.

[0054] 따라서 본 발명은 사면체의 수 ≒ 4(=사면체당 삼각형 수) ≒ 2만큼의 삼각형을 저장할 만큼의 공간을 메모리 풀에 잡아 두고 사용한다. 또한, 메모리 풀의 공간은 각 스레드들에게 분배되면, 각 스레드는 동기화없이 메모리 풀을 접근한다.

[0055] 본 발명은 제안하는 연결정보 추출 알고리즘을 두 개의 옥타코어 CPU(Intel Xeon Silver 4110, 2.10GHz) 및 384GB 메모리를 가진 윈도우즈(Windows) 시스템에서 VTK 8.2 프레임워크를 기반으로 시뮬레이션을 구현하였다.

[0056] 병렬처리 알고리즘은 OpenMP를 기반으로 구현하였으며, 하이퍼스레딩(hyper-threading) 기능을 이용하여 최대 32개 스레드를 사용하여 성능을 측정하였다.

[0057] 병렬처리 알고리즘은 동적 메모리 할당을 사용하는 버전(Ours w/o memory pool)과 메모리 풀을 사용(Ours)하는 두 가지 버전으로 구성하였다.

[0058] 벤치마크를 위해 본 발명은 서로 다른 크기를 가지는 세 가지 비정렬 격자 데이터에 대해 볼륨 렌더링을 수행하였으며(도 4 참조), 연결정보 추출 성능을 테스트 하였다. 각각의 비정렬 격자 데이터에 대한 정보는 하기의 표 1과 같다.

표 1

[0059]

	The number of primitives				Required memory for triangle list(s)	
	Vertex	Tetra	Face	Boundary face	VTK (GB)	Ours (GB)
Dataset 1	647,073	3,773,943	7,583,488	71,614	0.41	0.82
Dataset 2	7,320,994	41,257,368	83,385,432	1,741,392	4.41	9.01
Dataset 3	7,659,517	41,939,085	85,051,683	2,347,026	4.47	9.19

[0060] 도 5a 내지 도 5c는 본 발명이 제안한 병렬처리 알고리즘의 스레드 수 증가에 따른 성능 변화를 보여준다. 스레드 수를 증가시키며 본 발명의 병렬처리 알고리즘(Ours)의 성능을 측정한 결과, 모든 벤치마크 데이터에서 스레드 수에 따라 성능이 높아지는 경향성을 확인할 수 있다.

[0061] 상대적으로 크기가 작은 Dataset 1의 경우, 16개 스레드를 사용할 때 1개 스레드 사용 대비 5.27배 높은 성능을 보여주었으며, 32개 스레드(물리코어 16개) 사용 시에는 16개 스레드 사용대비 낮은 성능을 보여주었다. 이는 수행할 연산이 양이 많지 않아 병렬 처리 과정에서 발생하는 비용(메모리 풀 관리, 각 단계 사이의 동기화 등)이 상대적으로 크게 작용하기 때문이다. 하지만, 대용량 비정렬 격자 데이터인 Dataset 2와 3에서는 스레드 수 증가에 따라 지속적 성능 향상을 얻는다.

[0062] Dataset 2의 경우, 단일 스레드 사용대비 4개, 8개, 16개, 32개(물리코어 16개)의 스레드 사용에 대해 각각 3.4배, 6.1배, 8.0배, 8.9배 높은 성능을 보여준다. Dataset 3에서는 4개, 8개, 16개, 32개(물리코어 16개)의 스레드 사용에 대해 각각 3.3배, 5.6배, 7.5배, 8.5배 높은 성능을 보여주었다(표 2 참조).

표 2

[0063]

Dataset 1	# of threads	1	2	4	8	16	32
	Ours w/o mem. pool	8.42	4.75	2.94	1.81	1.66	1.80
	Ours	6.71	3.46	2.11	1.33	1.27	1.41
Dataset 2	# of threads	1	2	4	8	16	32
	Ours w/o mem. pool	216.04	118.43	61.33	36.85	28.63	24.74
	Ours	188.62	100.11	56.18	31.19	23.50	21.29
Dataset 3	# of threads	1	2	4	8	16	32
	Ours w/o mem. pool	239.75	142.44	72.21	43.09	32.57	28.18
	Ours	210.97	125.86	64.08	37.59	28.21	24.89

[0064] 이러한 확장성(scalability)은, 본 발명이 제안한 3-단계 알고리즘이 각 단계 내에서 동기화 없이 스레드들이 독립적으로 작업을 수행하기 때문에 얻을 수 있는 결과다.

[0065] 메모리 풀 활용 효율: 도 5a 내지 도 5c는 동적 메모리를 사용하는 경우와 스레드 마다 지역 메모리 풀을 사용하는 버전의 성능 비교를 보여준다. 메모리 풀을 사용하는 경우, 동적 메모리 할당 대비 최대 39% (평균 21%) 높은 성능을 보여준다.

[0066] 또한 모든 벤치마크 데이터에서 스레드 수와 무관하게 성능 향상을 얻는 것을 확인할 수 있다. 이러한 결과는 병렬 처리 과정에서 직렬화를 발생시키는 동적메모리 할당 과정을 제거함으로서 얻을 수 있는 결과다.

[0067] 공간적 지역성 개선 효과: 흥미로운 결과 중 하나는, 본 발명의 알고리즘을 사용하는 경우, 1개의 스레드만 사용한 경우에도 기존 직렬처리 연결정보 추출 알고리즘(VTK 구현) 대비 높은 성능을 보여준다는 것이다. Dataset 1, 2, 3에 대해 기존 직렬 알고리즘 대비 각각 1.4배, 5.3배, 9.8배 높은 성능을 보여준다(Table 4).

[0068] 이는 본 발명이 제안하는 3-단계 알고리즘이 중복 검사 시 기존 직렬처리 연결정보 추출 알고리즘 방법 대비 높은 공간적 지역성(spatial locality)을 갖기 때문이다. 사면체로부터 삼각형 정보를 생성할 때 바로 중복검사를 수행하는 기존의 알고리즘의 경우, 생성되는 삼각형들 순서와 해시 키의 순서에 규칙성이 없다. 즉, 중복 검사를 위해 접근하는 해시 테이블의 영역이 불규칙하며, 이러한 불규칙한 메모리 접근은 높은 캐시(cache) 활용 효율(예, hit ratio)을 얻기 힘들다.

[0069] 반면, 본 발명의 3-단계 알고리즘은 해시 키 단위로 스레드들에게 작업을 할당 하며, 각 스레드는 각 해시 키를 하나씩 처리한다. 즉, 같은 해시 키를 가지는 삼각형들 사이의 중복 검사를 수행하며, 해당 해시 키의 연결 리스트를 반복적으로 참조한다. 이처럼 메모리 접근에 있어 높은 공간적 지역성을 가지고 캐시 활용 효율을 높임으로서, 단일 스레드를 사용하는 경우에도 기존 직렬처리 연결정보 추출 알고리즘 대비 높은 성능 향상을 보여주었다.

[0070] 그 결과, 본 발명이 제안하는 병렬처리 알고리즘은 두 개의 옥타코어 CPU를 사용하여, 기존의 VTK의 직렬 알고리즘 대비 Dataset 1, 2, 3에 대해 각각 최대 6.6배, 47.3배, 82.9배 높은 성능을 보여준다. 이러한 결과는 본 발명 결과의 우수성을 보여준다. 또한, 데이터의 크기가 크고 연결 정보 추출의 시간이 큰 경우 더 높은 성능 향상을 보여준다. 즉, 이러한 결과는 본 발명이 제안하는 알고리즘의 대용량 비정렬 격자 데이터 처리에 대한 적합성을 증명한다.

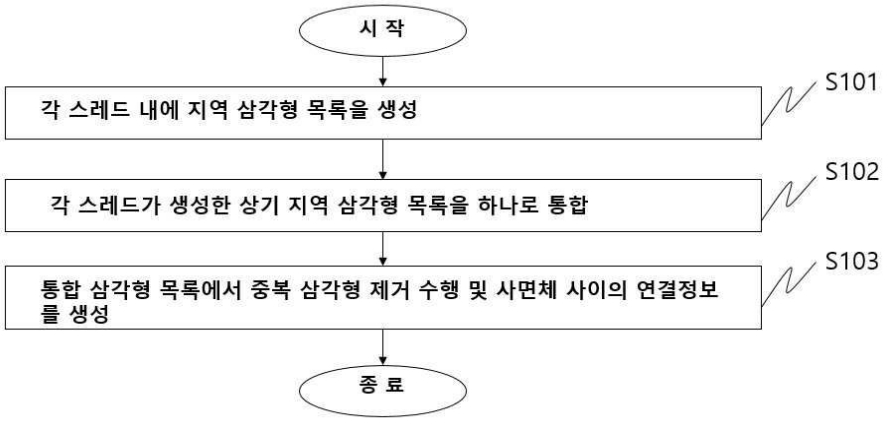
[0071] 나아가 본 발명의 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법은 컴퓨터와의 결합을 통해 실행시키기

위한 저장매체에 저장된 컴퓨터 프로그램으로 구현되거나, 이 컴퓨터 프로그램이 동작하는 컴퓨터 하드웨어에 탑재되는 모듈 형태로 구현될 수 있다.

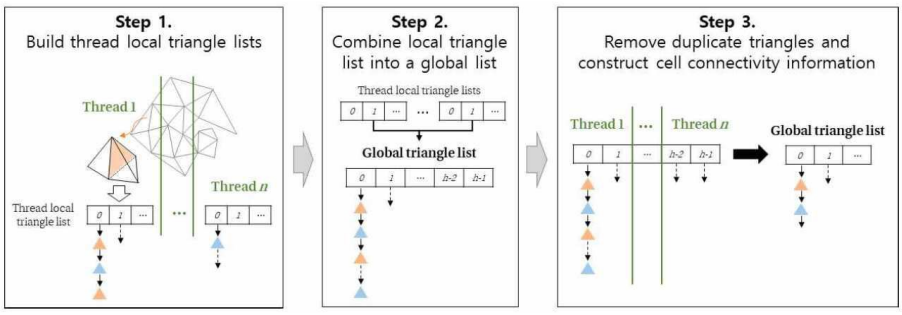
[0072] 또한, 본 발명의 일 실시예에 따른 병렬 처리에 의한 연결정보 추출 알고리즘 수행 방법의 동작 방법은, 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리, SSD(Solid State Drive) 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

도면

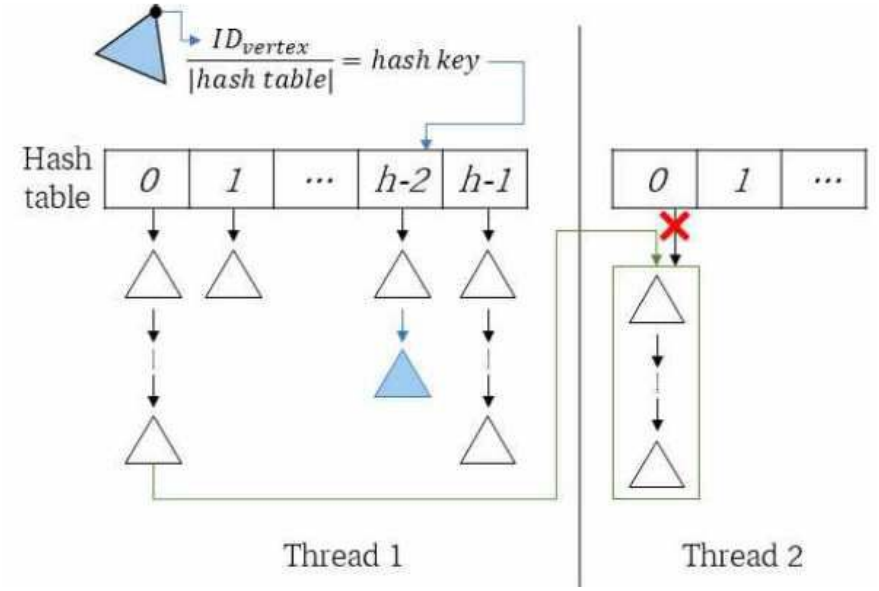
도면1



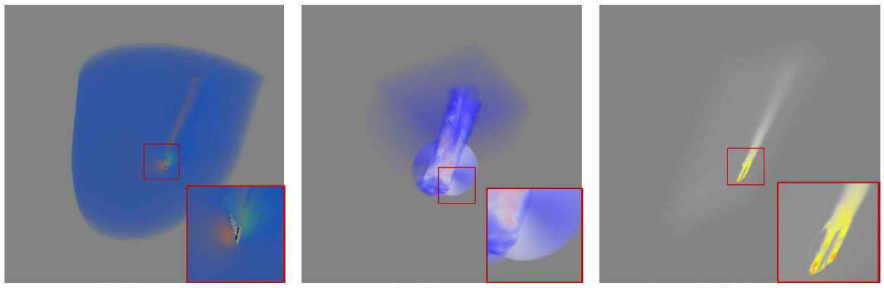
도면2



도면3

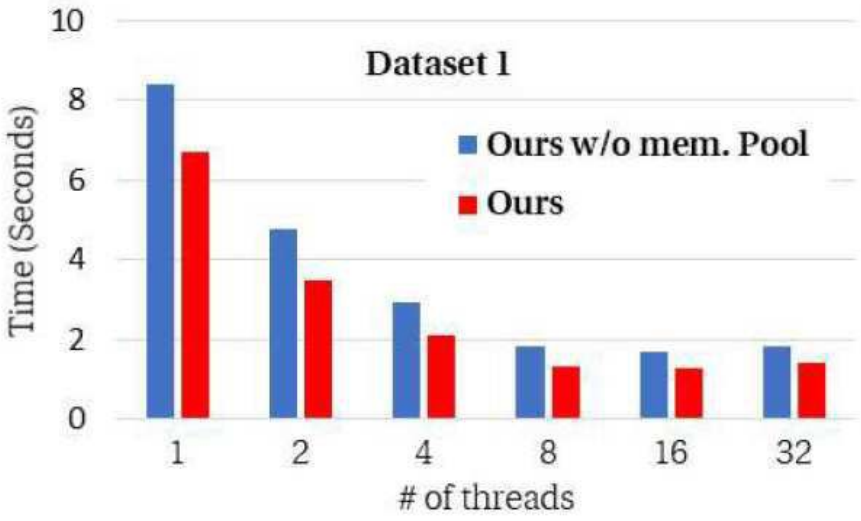


도면4

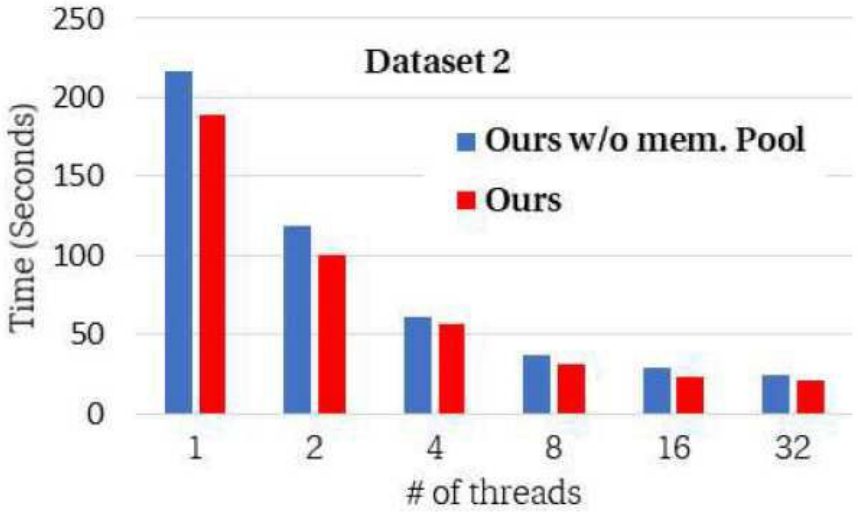


(a) Dataset 1 (b) Dataset 2 (C) Dataset 3

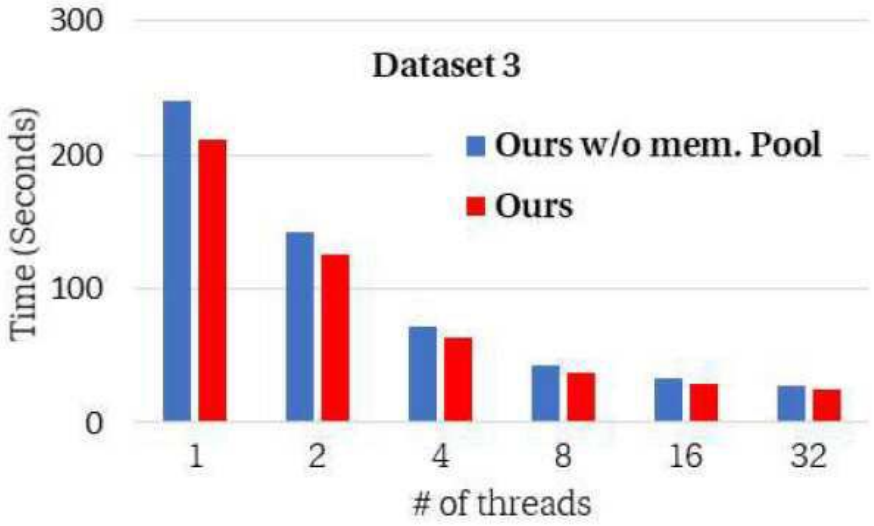
도면5a



도면5b


Dataset 2
Ours w/o mem. Pool
Ours
Time (Seconds)
250
200
150
100
50
0
1
2
4
8
16
32
# of threads

도면5c


Dataset 3
Ours w/o mem. Pool
Ours
Time (Seconds)
300
200
100
0
1
2
4
8
16
32
# of threads